Budgeted Matching and Budgeted Matroid Intersection via the Gasoline Puzzle

Andre Berger[*]   Vincenzo Bonifaci[*]   Fabrizio Grandoni[†]   Guido Schäfer[*]

Abstract

Many polynomial-time solvable combinatorial optimization problems become NP-hard if an additional complicating constraint is added to restrict the set of feasible solutions. In this paper, we consider two such problems, namely maximum-weight matching and maximum-weight matroid intersection with one additional budget constraint. We present the first polynomial-time approximation schemes for these problems.

Similarly to other approaches for related problems, our schemes compute two solutions to the Lagrangian relaxation of the problem and patch them together. However, due to the richer combinatorial structure of the problems considered here, standard patching techniques do not apply. To circumvent this problem, we crucially exploit the adjacency relations on the solution polytope and, somewhat surprisingly, the solution to an old combinatorial puzzle.

---

[*]Technische Universität Berlin, Institut für Mathematik, Germany, Email: {berger, bonifaci, schaefer}@math.tu-berlin.de

[†]Dipartimento di Informatica, Sistemi e Produzione, Università di Roma "Tor Vergata", Italy, Email: grandoni@disp.uniroma2.it

1 Introduction

Many combinatorial optimization problems can be formulated as follows. We are given a (finite) set $\mathscr{F}$ of feasible solutions and a weight function $w : \mathscr{F} \to \mathbb{Q}$ that assigns a weight $w(S)$ to every feasible solution $S \in \mathscr{F}$. An *optimization problem* Π asks for the computation of a feasible solution $S^* \in \mathscr{F}$ of maximum weight opt_Π, i.e.,

$$\text{opt}_\Pi := \text{maximize } w(S) \text{ subject to } S \in \mathscr{F}. \tag{Π}$$

In this paper, we are interested in solving such optimization problems if the set of feasible solutions is further constrained by a single *budget constraint*. More precisely, we are additionally given a non-negative cost function $c : \mathscr{F} \to \mathbb{Q}^+$ that specifies a cost $c(S)$ for every feasible solution $S \in \mathscr{F}$ and a non-negative budget $B \in \mathbb{Q}^+$. The *budgeted optimization problem* $\bar{\Pi}$ of the above problem Π can then be formulated as follows:

$$\text{opt} := \text{maximize } w(S) \text{ subject to } S \in \mathscr{F}, \ c(S) \leq B. \tag{$\bar{\Pi}$}$$

Even if the original optimization problem Π is polynomial-time solvable, adding a budget constraint typically renders the budgeted optimization problem $\bar{\Pi}$ NP-hard. Problems that fall into this class are, for example, the constrained shortest path problem [2], the constrained minimum spanning tree problem [1], and the constrained minimum arborescence problem [6].

In this paper, we study the budgeted version of two fundamental optimization problems, namely maximum-weight matching and maximum-weight matroid intersection.

In the *budgeted matching problem*, we are given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}$ and edge costs $c : E \to \mathbb{Q}^+$, and a budget $B \in \mathbb{Q}^+$. The set $\mathscr{F}$ of feasible solutions corresponds to the set of all matchings in G. The weight (cost) of a matching $M \in \mathscr{F}$ is simply given by the total weight (cost) of all edges in M.

In the *budgeted matroid intersection problem*, we are given two matroids $\mathscr{M}_1 = (E, \mathscr{F}_1)$ and $\mathscr{M}_2 = (E, \mathscr{F}_2)$ on a common ground set of elements E. (Formal definitions will be given in Section 2.) Moreover, we are given element weights $w : E \to \mathbb{Q}$, element costs $c : E \to \mathbb{Q}^+$, and a budget $B \in \mathbb{Q}^+$. The set of all feasible solutions $\mathscr{F}$ is defined by the intersection of $\mathscr{M}_1$ and $\mathscr{M}_2$. $\bar{\Pi}$ then corresponds to the computation of a common independent set $X \in \mathscr{F}_1 \cap \mathscr{F}_2$ of maximum weight $w(X) := \sum_{e \in X} w(e)$ that satisfies $c(X) := \sum_{e \in X} c(e) \leq B$. Problems that can be formulated as the intersection of two matroids are, for example, matchings in bipartite graphs, arborescences in directed graphs, spanning forests in undirected graphs, etc.

A special case of both budgeted matching and budgeted matroid intersection is the budgeted matching problem on bipartite graphs. This problem is NP-hard by a simple reduction from the knapsack problem. We remark that the unbudgeted versions of the two problems can be solved in polynomial-time (see, e.g., [20]).

Our Contribution. We give the first polynomial-time approximation schemes (PTAS) for the budgeted matching and the budgeted matroid intersection problem. For a given input parameter $\varepsilon > 0$, our algorithms compute a $(1 - \varepsilon)$-approximate solution in time $O(m^{O(1/\varepsilon)})$, where m is the number of edges in the graph or the number of elements in the ground set, respectively.

The basic structure of our polynomial-time approximation schemes resembles similar approaches for related budgeted optimization problems [18]. By dualizing the budget constraint of $\bar{\Pi}$ and lifting it into the objective function, we obtain for any $\lambda \geq 0$ the Lagrangian relaxation $\text{LR}(\lambda)$.

$$z(\lambda) := \text{maximize } \big(w(S) + \lambda(B - c(S)) \big) \text{ subject to } S \in \mathscr{F}. \tag{$\text{LR}(\lambda)$}$$

1

Note that the relaxed problem $LR(\lambda)$ is equivalent to the optimization problem Π with modified *Lagrangian weights* $w_\lambda(e) := w(e) - \lambda c(e)$ for all $e \in E$. Since the unbudgeted problem Π is polynomial-time solvable, we can compute the optimal *Lagrangian multiplier* $\lambda^* := \arg\min_{\lambda \geq 0} z(\lambda)$ and two optimal solutions S_1 and S_2 to $LR(\lambda^*)$ such that $c(S_1) \leq B \leq c(S_2)$. (Details will be given in Section 2.) The idea now is to *patch* S_1 and S_2 together to obtain a feasible solution S for $\bar{\Pi}$ whose weight $w(S)$ is at least $(1 - \varepsilon)\text{opt}$. Our patching consists of two phases: an *exchange phase* and an *augmentation phase*.

Exchange Phase: Consider the polytope induced by the feasible solutions $\mathscr{F}$ to the unbudgeted problem Π and let F be the face given by the solutions of maximum Lagrangian weight. This face contains both S_1 and S_2. In the first phase, we iteratively replace either S_1 or S_2 with another vertex on F, preserving the invariant $c(S_1) \leq B \leq c(S_2)$, until we end up with two adjacent solutions. Note that both solutions have objective value $z(\lambda^*) \geq \text{opt}$. However, with respect to their original weights, we can only infer that $w(S_i) = z(\lambda^*) - \lambda^*(B - c(S_i))$. That is, we cannot hope to use these solutions directly: S_1 is a feasible solution for $\bar{\Pi}$ but its weight $w(S_1)$ might be arbitrarily far from opt. In contrast, S_2 has weight $w(S_2) \geq \text{opt}$, but is infeasible.

Augmentation Phase: In this phase, we exploit the properties of adjacent solutions in the solution polytope. For matchings it is known that two solutions are adjacent in the matching polytope if and only if their symmetric difference is an alternating cycle or path X. Analogously, two adjacent extreme points in the common basis polytope of two matroids can be characterized by a proper alternating cycle X in the corresponding exchangeability graph [4, 9]. The idea is to patch S_1 according to a proper subpath X' of X. This subpath X' guarantees that the Lagrangian weight of S_1 does not decrease too much, while at the same time the gap between the budget and the cost of S_1 (and hence also the gap between $w(S_1)$ and $z(\lambda^*)$) is reduced. This way we obtain a feasible solution S whose weight differs from opt by at most the weight of two edges (elements).

Of course, constructing such a solution S alone is not sufficient to obtain a PTAS. (The maximum weight of an edge (element) might be comparable to the weight of an optimum solution). However, this problem can be easily overcome by guessing the edges (elements) of largest weight in the optimum solution in a preliminary step.

Surprisingly, the key ingredient that enables us to prove that there always exists a good patching subpath stems from an old combinatorial puzzle which we quote from the book by Lovász [10, Problem 3.21]. We leave the proof as an exercise to the reader.

> "Along a speed track there are some gas-stations. The total amount of gasoline available in them is equal to what our car (which has a very large tank) needs for going around the track. Prove that there is a gas-station such that if we start there with an empty tank, we shall be able to go around the track without running out of gasoline."

Related Work. For the budgeted matching problem there is an optimal algorithm if the costs are uniform. This problem is equivalent to finding a maximum-weight matching that consists of at most B edges, which can be solved by a reduction to perfect matching. Not much is known for the budgeted matching problem with general edge costs, besides that it is NP-hard. Naor et al. [15] proposed an FPTAS for an even more general class of problems, which contains the budgeted matching problem considered here as special case. However, personal communication [14] revealed that unfortunately the stated result [15, Theorem 2.2] is incorrect. To the best of our knowledge, the budgeted version of the maximum-weight matroid intersection problem has not been considered before.

Budgeted versions of polynomial-time solvable optimization problems have been studied extensively. The most known ones are probably the constrained shortest path problem and the constrained minimum spanning tree problem. Finding a shortest s,t-path P (with respect to weight) between two vertices s and t in a directed graph with edge weights and edge costs such that the total cost of P is at most B appears as an NP-hard problem already in the book by Garey and Johnson [5]. Similarly, finding a minimum weight spanning tree whose total cost is at most some specified value is NP-hard as well [1].

Goemans and Ravi [18] obtain a PTAS for the constrained minimum spanning tree problem by using an approach which resembles our exchange phase. Starting from two spanning trees obtained from the Lagrangian relaxation, they walk along the optimal face (with respect to the Lagrangian weights) of the spanning tree polytope until they end up with two adjacent solutions S_1 and S_2 with $c(S_1) \leq B \leq c(S_2)$. In this polytope, two spanning trees are adjacent if and only if their symmetric difference consists of just two edges. Therefore, the final solution S_1 is a feasible spanning tree whose weight is away from the optimum by the weight of only one edge. In particular, once two such adjacent solutions have been found there is no need for an additional augmentation phase, which is instead crucial for matchings and matroid intersections. The PTAS by Goemans and Ravi [18] also extends to the problem of finding a minimum-weight basis in a matroid subject to a budget constraint.

Hassin and Levin [7] later improved the result of Goemans and Ravi and obtained an EPTAS for the constrained minimum spanning tree problem. A fully polynomial bicriteria approximation scheme for the problem has been found by Hong et al. [8]. However, the question whether there exists a fully polynomial time approximation scheme to the constrained minimum spanning tree problem is open.

Finding constrained minimum arborescences in directed graphs is NP-hard as well. Guignard and Rosenwein [6] apply Langrangian relaxatian to solve it to optimality (though not in polynomial time). Previous work on budgeted optimization problems also includes results on budgeted scheduling [21] and bicriteria results for several budgeted network design problems [11].

All problems mentioned above can be interpreted as bicriteria optimization problems with a min-min objective, i.e., where the goal is to compute a solutions that minimizes the objective value and whose cost stays below a given budget. In contrast, in our work we consider max-min bicriteria problems.

Organization of the Paper. The paper is structured as follows. In Section 2, we give some prerequisites on matroids and Lagrangian relaxation. We then present the PTAS for the budgeted matching problem in Section 3. The PTAS for the budgeted matroid intersection problem is the subject of Section 4. In Section 5 we discuss some open problems.

2 Preliminaries

2.1 Matroids

Let E be a set of elements and $\mathscr{F} \subseteq 2^E$ be a non-empty set of subsets of E. Then $\mathscr{M} = (E, \mathscr{F})$ is a *matroid* if the following holds:

(a) If $I \in \mathscr{F}$ and $J \subseteq I$, then $J \in \mathscr{F}$.

(b) For every $I, J \in \mathscr{F}$, $|I| = |J|$, for every $x \in I$ there is $y \in J$ such that $I \setminus \{x\} \cup \{y\} \in \mathscr{F}$.

The elements of $\mathscr{F}$ are called *independent sets*. An independent set X is a *basis* of $\mathscr{M}$ if for every $x \in E \setminus X$, $X \cup \{x\} \notin \mathscr{F}$. We assume that $\mathscr{F}$ is represented implicitly by an oracle: for any given

$I \subseteq E$, this oracle determines whether $I \in \mathscr{F}$ or not. In the running time analysis, each query to the oracle is assumed to take constant time. It is not hard to show that matroids have the following properties (see e.g. [20] and references therein).

Lemma 1. *For any given matroid $\mathscr{M} = (E, \mathscr{F})$:*

1. *(deletion) For every $E_0 \subseteq E$, $\mathscr{M} - E_0 := (E', \mathscr{F}')$ is a matroid, where $E' := E \setminus E_0$ and $\mathscr{F}' := \{X \in \mathscr{F} : X \cap E_0 = \emptyset\}$.*

2. *(contraction) For every $E_0 \in \mathscr{F}$, $\mathscr{M}/E_0 := (E', \mathscr{F}')$ is a matroid, where $E' := E \setminus E_0$ and $\mathscr{F}' := \{X \subseteq E \setminus E_0 : X \cup E_0 \in \mathscr{F}\}$.*

3. *(truncation) For every $q \in \mathbb{N}$, $\mathscr{M}^q := (E, \mathscr{F}^q)$ is a matroid, where $\mathscr{F}^q := \{X \in \mathscr{F} : |X| \leq q\}$.*

4. *(extension) For every set D, $D \cap E = \emptyset$, $\mathscr{M} + D := (E', \mathscr{F}')$ is a matroid, where $E' := E \cup D$ and $\mathscr{F}' := \{X \subseteq E \cup D : X \cap E \in \mathscr{F}\}$.*

Observe that an oracle for the original matroid implicitly defines an oracle for all the derived matroids above. Given $X \in \mathscr{F}$ and $Y \subseteq E$, the *exchangeability graph* of $\mathscr{M}$ with respect to X and Y is the bipartite graph $\mathbf{ex}_{\mathscr{M}}(X,Y) := (X \setminus Y, Y \setminus X; H)$ with edge set $H = \{(x,y) : x \in X \setminus Y, y \in Y \setminus X, X \setminus \{x\} \cup \{y\} \in \mathscr{F}\}$.

Lemma 2 ([9]). *(Exchangeability Lemma) Given $X \in \mathscr{F}$ and $Y \subseteq E$, if $\mathbf{ex}_{\mathscr{M}}(X,Y)$ has a unique perfect matching, then $Y \in \mathscr{F}$.*

The *intersection* of two matroids $\mathscr{M}_1 = (E, \mathscr{F}_1)$ and $\mathscr{M}_2 = (E, \mathscr{F}_1)$ over the same ground set E is the pair $\mathscr{M} = (E, \mathscr{F}_1 \cap \mathscr{F}_2)$. We remark that the intersection of two matroids might not be a matroid, while every matroid $\mathscr{M} = (E, \mathscr{F})$ is the intersection of itself with the trivial matroid $(E, 2^E)$. Lemma 1 can be naturally extended to matroid intersections. For example, for a given matroid intersection $(E, \mathscr{F}_1 \cap \mathscr{F}_2)$, by Lemma 1.3 $(E, \mathscr{F}_1^q \cap \mathscr{F}_2^q)$ is still the intersection of two matroids, for any $q \in \mathbb{N}$.

Given two matroids $\mathscr{M}_1 = (E, \mathscr{F}_1)$ and $\mathscr{M}_2 = (E, \mathscr{F}_1)$, the *common basis polytope* of $\mathscr{M}_1$ and $\mathscr{M}_2$ is the convex hull of the characteristic vectors of the common bases. We say that two common bases $X, Y \in \mathscr{F}_1 \cap \mathscr{F}_2$ are *adjacent* if their characteristic vectors are adjacent extreme points in the common basis polytope of $\mathscr{M}_1$ and $\mathscr{M}_2$.

2.2 Lagrangian Relaxation

We briefly review the *Lagrangian relaxation* approach; for a more detailed exposition, the reader is referred to [16]. The Lagrangian relaxation of the budgeted optimization problem $\bar{\Pi}$ is given by:

$$z(\lambda) := \text{maximize } \big(w(S) + \lambda(B - c(S))\big) \text{ subject to } S \in \mathscr{F} \qquad (\text{LR}(\lambda))$$

For any value of $\lambda \geq 0$, the optimal solution to $\text{LR}(\lambda)$ gives an upper bound on the optimal solution of the original budgeted problem, because any feasible solution satisfies $\sum_{e \in S} c(e) \leq B$. The Lagrangian relaxation problem is to find the best such upper bound, i.e. to determine λ^* such that $z(\lambda^*) = \min_{\lambda \geq 0} z(\lambda)$. This can be done in polynomial time whenever $\text{LR}(\lambda)$ is solvable in polynomial time [19, Theorem 24.3]. In our case, since there are combinatorial algorithms for weighted matching and weighted matroid intersection [20], we can even obtain λ^* in strongly polynomial time by using Megiddo's parametric search technique [12]. Indeed, for any fixed feasible solution, the value of the Lagrangian relaxation is a linear function of λ, so that $\text{LR}(\lambda)$ is the maximum of a set of linear functions, i.e. a piecewise-linear convex function. Finding the minimum of such a piecewise-linear convex function is precisely what is achieved by parametric search.

The idea behind Megiddo's technique is that, even though we do not know λ^*, we can simulate the algorithm solving $LR(\lambda^*)$ and at the same time discover λ^*. Towards this end, for each $e \in E$ the value $w_{\lambda^*}(e) := w(e) - \lambda^* c(e)$ will be manipulated symbolically as a linear function of the form $a + \lambda^* b$. In the simulated algorithm, which is combinatorial, these linear functions might be added together to create more linear functions, but at most a polynomial number of such functions will be used overall. Whenever the simulated algorithm asks for a comparison between some $a + \lambda^* b$ and some $a' + \lambda^* b'$, we compute the critical λ for which $a + \lambda b = a' + \lambda b'$. To correctly perform the comparison and resume the simulation of the algorithm, it is enough to know whether λ is smaller or larger than λ^*. But this can be discovered by solving one more Lagrangian subproblem, this time with weight function $w - \lambda c$: if the corresponding solution costs more than B, then $\lambda < \lambda^*$, and viceversa if the cost is larger than B. At the end of the simulation, the output of the algorithm can be used to determine λ^* explicitly. Finally, λ^* can be used to compute two solutions S_1, S_2 such that:

1. Both S_1 and S_2 are optimal with respect to the weight function $w_{\lambda^*}(e) := w(e) - \lambda^* c(e)$, $e \in E$;

2. $c(S_1) \leq B \leq c(S_2)$.

These two solutions can be obtained by solving the relaxed problems $LR(\lambda^* + \varepsilon)$ and $LR(\lambda^* - \varepsilon)$, respectively, for a sufficiently small $\varepsilon > 0$. Indeed, even without knowing how small ε has to be, they can be obtained by simulating again the algorithm and resolving the comparisons accordingly.

2.3 The Gasoline Puzzle

One crucial ingredient in our patching procedure is the solution to the puzzle cited in the introduction. We state it more formally in the following lemma.

Lemma 3. *(Gasoline Lemma) Given a sequence of k real values $a_0, a_1, \ldots a_{k-1}$ of total value $\sum_{j=0}^{k-1} a_j = 0$, there is an index $i \in \{0, 1, \ldots, k-1\}$ such that, for any $0 \leq h \leq k-1$, $\sum_{j=i}^{i+h} a_{j \pmod k} \geq 0$.*

3 A PTAS for the Budgeted Matching Problem

In this section, we present our PTAS for the budgeted matching problem. Suppose we are given a budgeted matching instance $I := (G, w, c, B)$. Let n and m refer to the number of nodes and edges in G, respectively. Moreover, we define $w_{\max} := \max_{e \in E} w(e)$ as the largest edge weight in I. Throughout this section, opt refers to the weight of an optimal solution M^* for I. In order to prove that there exists a PTAS, we proceed in two steps: First we prove that there is an algorithm to compute a feasible solution of weight at least $opt - 2w_{\max}$. The Gasoline Lemma will play a crucial role in this proof. The claimed PTAS is then obtained by guessing the edges of largest weight in M^* in a preliminary phase and applying the algorithm above.

Lemma 4. *There is a polynomial-time algorithm to compute a solution M to the budgeted matching problem of weight $w(M) \geq opt - 2w_{\max}$.*

Proof. As described in Section 2, we first compute the optimal Lagrangian multiplier $\lambda > 0$ and two matchings M_1 and M_2 of maximum Lagrangian weight $w_\lambda(M_1) = w_\lambda(M_2)$ and satisfying $c(M_1) \leq B \leq c(M_2)$. Observe that for $i \in \{1, 2\}$ we have that

$$w_\lambda(M_i) + \lambda B \geq w_\lambda(M^*) + \lambda B \geq w_\lambda(M^*) + \lambda c(M^*) = opt. \tag{1}$$
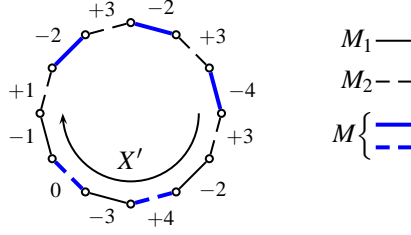
Figure 1: The construction used in Lemma 4. Each edge x_i is labeled with the value a_i.

We next show how to extract from $M_1 \cup M_2$ a matching M with the desired properties in polynomial-time. Consider the symmetric difference $M' = M_1 \oplus M_2$. Recall that $M' \subseteq M_1 \cup M_2$ consists of a disjoint union of paths $\mathscr{P}$ and cycles $\mathscr{C}$. We apply the following procedure until eventually $|\mathscr{P} \cup \mathscr{C}| \leq 1$: Take some $X \in \mathscr{P} \cup \mathscr{C}$ and let $A := M_1 \oplus X$. If $c(A) \leq B$ replace M_1 by A. Otherwise replace M_2 by A. Observe that in each step, the cardinality of $M_1 \cap M_2$ increases by at least one; hence this procedure terminates after at most $O(n)$ steps. Moreover, by the optimality of M_1 and M_2, the Lagrangian weight of the two matchings does not change during the process.

If at the end of this procedure $c(M_i) = B$ for some $i \in \{1, 2\}$, we are done: M_i is a feasible solution to the budgeted matching problem and

$$w(M_i) = w_\lambda(M_i) + \lambda c(M_i) = w_\lambda(M_i) + \lambda B \geq \text{opt}.$$

Otherwise, $M_1 \oplus M_2$ consists of a unique path or cycle $X = (x_0, x_1, \ldots, x_{k-1})$ such that $c(M_1 \oplus X) = c(M_2) > B > c(M_1)$. Consider the sequence

$$a_0 = \delta(x_0) w_\lambda(x_0), \quad a_1 = \delta(x_1) w_\lambda(x_1), \quad \ldots \quad a_{k-1} = \delta(x_{k-1}) w_\lambda(x_{k-1}),$$

where $\delta(x_i) = 1$ if $x_i \in M_2$ and $\delta(x_i) = -1$ otherwise. This sequence has total value zero, because of the optimality of M_1 and M_2. By the Gasoline Lemma, there must exist an edge x_i, $i \in \{0, 1, \ldots, k-1\}$, of X such that for any cyclic subsequence $X' = (x_i, x_{(i+1) \pmod k}, \ldots, x_{(i+h) \pmod k})$,

$$0 \leq \sum_{j=i}^{i+h} a_j \pmod k = \sum_{e \in X' \cap M_2} w_\lambda(e) - \sum_{e \in X' \cap M_1} w_\lambda(e). \tag{2}$$

Consider the longest such subsequence X' satisfying $c(M_1 \oplus X') \leq B$. Let $e_1 = x_i$ and $e_2 = x_{(i+h) \pmod k}$ be the endpoints of X' (see Figure 1). Note that by the maximality of X' and by the non-negativity of the edge costs, either $e_2 \in M_1$ or X is a path and e_2 its last edge. In both cases, $M := (M_1 \oplus X') \setminus \{e_1\}$ is a matching (while $M_1 \oplus X'$ might not be a matching if $e_1 \in M_2$). Moreover, $c(M) = c(M_1 \oplus X') - c(e_1) \leq c(M_1 \oplus X') \leq B$. That is, M is a feasible solution to the budgeted matching problem.

It remains to lower bound the weight of M. We have

$$w(M_1 \oplus X') = w_\lambda(M_1 \oplus X') + \lambda c(M_1 \oplus X') = w_\lambda(M_1 \oplus X') + \lambda B - \lambda (B - c(M_1 \oplus X'))$$
$$\geq w_\lambda(M_1) + \lambda B - \lambda (B - c(M_1 \oplus X')) \geq \text{opt} - \lambda (B - c(M_1 \oplus X')),$$

where the first inequality follows from (2) and the second inequality follows from (1).

Let $e_3 = x_{(i+h+1) \pmod k}$. The maximality of X' implies that $c(e_3) > B - c(M_1 \oplus X') \geq 0$. Moreover, by the optimality of M_1 and M_2, $0 \leq w_\lambda(e_3) = w(e_3) - \lambda c(e_3)$. Altogether $\lambda (B - c(M_1 \oplus X')) \leq \lambda c(e_3) \leq w(e_3)$ and hence $w(M_1 \oplus X') \geq \text{opt} - w(e_3)$. We can thus conclude that

$$w(M) = w(M_1 \oplus X') - w(e_1) \geq \text{opt} - w(e_3) - w(e_1) \geq \text{opt} - 2w_{\max}. \qquad \square$$

Theorem 1. *There is a PTAS for the budgeted matching problem.*

Proof. Let $\varepsilon \in (0,1)$ be a given constant. Assume that the optimum matching M^* contains at least $p := \lceil 2/\varepsilon \rceil$ edges. (Otherwise the problem can be solved optimally by brute force.) Consider the following algorithm. Initially, we guess the p heaviest (with respect to weights) edges M_H^* of M^*. Then we remove from the graph G the edges in M_H^*, all edges incident to M_H^*, and all edges of weight larger than the smallest weight in M_H^*. We also decrease the budget by $c(M_H^*)$. Let I' be the resulting budgeted matching instance. Note that the maximum weight of an edge in I' is $w'_{\max} \le w(M_H^*)/p \le \varepsilon M_H^*/2$. Moreover, $M_L^* := M^* \setminus M_H^*$ is an optimum solution for I'. We compute a matching M' for I' using the algorithm described in the proof of Lemma 4. Eventually, we output the feasible solution $M := M_H^* \cup M'$.

The algorithm above has running time $O(m^{p+O(1)}) = O(m^{O(1/\varepsilon)})$, where the m^p factor comes from the guessing of M_H^*. By Lemma 4, $w(M') \ge w(M_L^*) - 2w'_{\max}$. It follows that

$$w(M) = w(M_H^*) + w(M') \ge w(M_H^*) + w(M_L^*) - 2w'_{\max} \ge w(M^*) - \varepsilon w(M_H^*) \ge (1-\varepsilon)w(M^*).$$

$\square$

4 A PTAS for the Budgeted Matroid Intersection Problem

In this section we will develop a PTAS for the budgeted matroid intersection problem. As in the PTAS for the budgeted matching problem, we will first show how to find a feasible common independent set of two matroids $\mathscr{M}_1 = (E, \mathscr{F}_1)$ and $\mathscr{M}_2 = (E, \mathscr{F}_2)$ of weight at least $\mathrm{opt} - 2w_{\max}$, where $w_{\max}$ is the weight of the heaviest element. The PTAS will then follow similarly as in the previous section.

Like in the matching case, we initially use Megiddo's parametric search technique to obtain the optimal Lagrangian multiplier $\lambda \ge 0$ and two solutions $X, Y \in \mathscr{F}_1 \cap \mathscr{F}_2$, $c(X) \le B \le c(Y)$, that are optimal with respect to the Lagrangian weights $w_\lambda(e) = w(e) - \lambda c(e)$, $e \in E$. Notice that neither X nor Y will contain any element e such that $w_\lambda(e) < 0$. Furthermore, both solutions can be used to derive upper bounds on the optimum solution. In fact, let I^* be the optimum solution, of weight $\mathrm{opt} = w(I^*)$. For $Z \in \{X, Y\}$,

$$w_\lambda(Z) + \lambda B \ge w_\lambda(I^*) + \lambda B \ge w_\lambda(I^*) + \lambda c(I^*) = \mathrm{opt}. \tag{3}$$

If X and Y have different cardinalities, say $|X| < |Y|$, we extend $\mathscr{M}_1$ and $\mathscr{M}_2$ according to Lemma 1.4 by adding $|Y| - |X|$ dummy elements D of weight and cost zero, and then we replace X by $X \cup D$. (Dummy elements will be discarded when the final solution is returned.) Of course, this does not modify the weight of the optimum solution nor the weight and cost of X. Finally, using Lemma 1.3 we truncate the two matroids to $q := |X| = |Y|$. The solutions X and Y will now be maximum-weight common bases of each one of the two truncated matroids.

In the following, we will show how to derive from X and Y a feasible solution of weight at least $\mathrm{opt} - 2w_{\max}$. This is done in two steps. First (Section 4.1), we extract from $X \cup Y$ two adjacent common bases, one below and the other over the budget, with the same (optimal) Lagrangian weight of X and Y. Then (Section 4.2) we apply the Gasoline Lemma to a proper auxiliary graph to compute the desired approximate solution.

4.1 Finding Adjacent Common Bases

The following lemma characterizes two adjacent common bases in the common basis polytope of two matroids.

Lemma 5 ([4,9]). *Assume we have two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$, $\mathcal{M}_2 = (E, \mathcal{F}_2)$ and two common bases $X, Y \in \mathcal{F}_1 \cap \mathcal{F}_2$. Then X and Y are adjacent extreme points in the common basis polytope if and only if the following conditions hold:*

1. *The exchangeability graph $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$ has a unique perfect matching M_1.*

2. *The exchangeability graph $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$ has a unique perfect matching M_2.*

3. *The union $M_1 \cup M_2$ forms a connected cycle.*

The following corollary of Lemma 5 will help us to deal with contracted matroids.

Corollary 1. *Let $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$ be two matroids. Moreover, let $Z \in \mathcal{F}_1 \cap \mathcal{F}_2$ and $Z \subseteq X \cap Y$. Then X and Y are adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$ if and only if $X \setminus Z$ and $Y \setminus Z$ are adjacent extreme points in the common basis polytope of $\mathcal{M}_1 / Z$ and $\mathcal{M}_2 / Z$.*

Proof. First note, that X is a basis of $\mathcal{M}_i$ if and only if $X \setminus Z$ is a basis of $\mathcal{M}_i / Z$ $(i = 1, 2)$ by Lemma 1.2. The same holds for Y. Moreover, as $Z \subseteq X \cap Y$, the exchangeability graphs $\mathbf{ex}_{\mathcal{M}_i}(X, Y)$ and $\mathbf{ex}_{\mathcal{M}_i / Z}(X \setminus Z, Y \setminus Z)$ $(i = 1, 2)$ are the same, since they are defined on the symmetric difference of X and Y. The claim then follows immediately from Lemma 5. $\square$

Remember that X and Y, are maximum-weight common bases of $\mathcal{M}_1$ and $\mathcal{M}_2$ with respect to the Lagrangian weights w_λ, and that $c(X) \leq B \leq c(Y)$. Since our solution will be a subset of $X \cup Y$, let us delete the elements $E' = E \setminus (X \cup Y)$ according to Lemma 1.1. In order to do a similar patching procedure as for the matching problem, we would like X and Y to be adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$. The following lemma will help us to find such two adjacent common bases which are also of maximum weight with respect to w_λ.

Lemma 6. *There is a polynomial-time algorithm that finds a third maximum-weight common basis A with respect to w_λ, such that $X \neq A \neq Y$ and $X \cap Y \subseteq A \subseteq X \cup Y$, or determines that no such basis exists.*

Proof. Let $Z = X \cap Y$. Without loss of generality, let $X \setminus Y = \{x_1, \ldots, x_r\}$ and $Y \setminus X = \{y_1, \ldots, y_r\}$. For $1 \leq i, j \leq r$ denote by $\mathcal{M}_1^{ij} = \mathcal{M}_1 / Z - \{x_i, y_j\}$ and $\mathcal{M}_2^{ij} = \mathcal{M}_2 / Z - \{x_i, y_j\}$ the matroids resulting from the contraction of Z (Lemma 1.2) and the deletion of x_i and y_j (Lemma 1.1).

Consider the following (polynomial-time) algorithm. For every $1 \leq i, j \leq r$ compute A_{ij}, a maximum-weight common basis of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$. If there is an A_{ij} satisfying $|A_{ij}| = r$ and $w_\lambda(A_{ij}) = w_\lambda(X \setminus Z)$, then $A = A_{ij} \cup Z$ is the desired third basis. In fact, A_{ij} is a common basis of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$, and since $|A| = |A_{ij}| + |Z| = |X|$, it is also a common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$. Also, $X \neq A \neq Y$ since x_i and y_j are not present in $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$.

If none of the A_{ij}'s satisfies $|A_{ij}| = r$ and $w_\lambda(A_{ij}) = w_\lambda(X \setminus Z)$, then no common basis A of $\mathcal{M}_1$ and $\mathcal{M}_2$ with the desired properties exists. In fact, assume by contradiction that there is such a third maximum-weight basis A. Choose i and j such that $x_i, y_j \notin A$. Note that such indices must exist since $X \neq A \neq Y$. Then $A \setminus Z$ is a common basis of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$. Hence $w_\lambda(A_{ij}) \geq w_\lambda(A \setminus Z)$, since A_{ij} is a maximum-weight such common basis. Moreover $|A_{ij}| = |A \setminus Z| = r$, and thus $A_{ij} \cup Z$ is a common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$, implying $w_\lambda(A_{ij} \cup Z) \leq w_\lambda(A)$. Hence $w_\lambda(A_{ij}) \leq w_\lambda(A \setminus Z)$. We can conclude that $w_\lambda(A_{ij}) = w_\lambda(A \setminus Z) = w_\lambda(X \setminus Z)$, which is a contradiction. $\square$

We can now apply Lemma 6 as follows. Until we find a third basis A, we replace X by A if $c(A) \leq B$, and Y by A otherwise. In either case, the cardinality of the intersection of the new X and Y has increased. Hence this process ends in at most $O(m)$ rounds.

At the end of the process, X and Y must be adjacent in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$. In fact, $X \setminus Y$ and $Y \setminus X$ are maximum-weight common bases of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$ and there is no other maximum-weight common basis A' of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$, as otherwise $A = A' \cup (X \cap Y)$ would have been found by the algorithm from Lemma 6. Now as $X \setminus Y$ and $Y \setminus X$ are the only two maximum-weight common bases, they must also be adjacent on the optimal face of the common basis polytope of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$. Therefore, by Corollary 1, X and Y are adjacent in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$.

4.2 Merging Adjacent Common Bases

Let X and Y be the two adjacent solutions obtained at the end of the process described in the previous section. Notice that, if either $c(X) = B$ or $c(Y) = B$, we obtain a feasible solution that is optimal also with respect to the original weights, in which case we can already stop. For this reason in the following we will assume that $c(S_1) < B < c(S_2)$. Without loss of generality, we also assume that $X \setminus Y = \{x_1, x_2, \ldots, x_r\}$ an $Y \setminus X = \{y_1, y_2, \ldots, y_r\}$.

Lemma 7. *Given X and Y with the properties above, there is a polynomial-time algorithm which computes a common independent set $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $c(X') \leq B$ and $w(X') \geq \text{opt} - 2w_{\max}$.*

Proof. We exploit again Lemma 5 to obtain two unique perfect matchings: $M_1 = \{x_1 y_1, \ldots, x_r y_r\}$ in $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$ and $M_2 = \{y_1 x_2, y_2 x_3, \ldots, y_r x_1\}$ in $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$. Let $(x_1, y_1, x_2, y_2, \ldots, x_r, y_r)$ be the corresponding connected cycle. Assign to the each edge $x_j y_j$ a weight $\delta_j := w_\lambda(y_j) - w_\lambda(x_j)$, and weight zero to the remaining edges. Clearly $\sum_{j=1}^{r} \delta_j = 0$, since X and Y have the same maximum Lagrangian weight. Hence, by the Gasoline Lemma, there must exist an edge of the cycle such that the partial sum of the δ-weights of each subpath starting at that edge is nonnegative. Without loss of generality, assume $x_1 y_1$ is such an edge. Thus for all $i \leq r$, $\sum_{j=1}^{i} \delta_j \geq 0$. Find the largest $k \leq r$ such that $c(X) + \sum_{j=1}^{k} (c(y_j) - c(x_j)) \leq B$. Since $c(Y) > B$, we have $k < r$ and by construction $c(X) + \sum_{j=1}^{k} (c(y_j) - c(x_j)) > B - c(y_{k+1}) + c(x_{k+1})$.

We now show that $X' := X \setminus \{x_1, \ldots, x_{k+1}\} \cup \{y_1, \ldots, y_k\}$ satisfies the claim. By the choice of k, $B - c_{\max} \leq B - c(y_{k+1}) < c(X') \leq B$, where $c_{\max} = \max_{e \in E} c(e)$. Also, since $\sum_{j=1}^{k} \delta_j \geq 0$, we have $w_\lambda(X') \geq w_\lambda(X) - w_\lambda(x_{k+1}) \geq w_\lambda(X) - w_{\max}$.

We next prove that $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$. Consider the set $X' \cup \{x_{k+1}\}$: its symmetric difference with X is the set $\{x_1, \ldots, x_k\} \cup \{y_1, \ldots, y_k\}$. Recall that $x_i y_i$ is an edge of M_1. Thus, for $i \leq k$, it is also an edge of $\mathbf{ex}_{\mathcal{M}_1}(X, X' \cup \{x_{k+1}\})$ so that this graph has a perfect matching. On the other hand this perfect matching must be unique, otherwise M_1 would not be unique in $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$. Thus by the Exchangeability Lemma $X' \cup \{x_{k+1}\} \in \mathcal{F}_1$.

Similarly, consider the set $X' \cup \{x_1\}$: its symmetric difference with X is the set $\{x_2, \ldots, x_{k+1}\} \cup \{y_1, \ldots, y_k\}$. For $i \leq k$, $y_i x_{i+1}$ is an edge of M_2. Thus $\mathbf{ex}_{\mathcal{M}_2}(X, X' \cup \{x_1\})$ has a perfect matching, and it has to be unique, otherwise M_2 would not be unique in $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$. Thus by the Exchangeability Lemma $X' \cup \{x_1\} \in \mathcal{F}_2$. We have thus shown that $X' \cup \{x_{k+1}\} \in \mathcal{F}_1$ and $X' \cup \{x_1\} \in \mathcal{F}_2$. As a consequence, $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$.

It remains to bound the weight of X':

$$\begin{aligned} w(X') &= w_\lambda(X') + \lambda c(X') = w_\lambda(X') + \lambda B - \lambda(B - c(X')) \\ &\geq w_\lambda(X) + \lambda B - w_{\max} - \lambda c(y_{k+1}) \geq w_\lambda(X) + \lambda B - 2w_{\max} \geq \text{opt} - 2w_{\max}. \end{aligned}$$

Above we used the fact that $w_\lambda(e) \geq 0$ for all $e \in Y$, so in particular $w_\lambda(y_{k+1}) = w(y_{k+1}) - \lambda c(y_{k+1}) \geq 0$, implying $w_{\max} \geq w(y_{k+1}) \geq \lambda c(y_{k+1})$. The last inequality follows from (3). $\square$

Theorem 2. *The budgeted matroid intersection problem admits a PTAS.*

Proof. Let $\varepsilon \in (0,1)$ be a given constant. Assume that the optimum solution contains at least $p := \lceil 2/\varepsilon \rceil$ elements (otherwise the problem can be solved optimally by brute force). We first guess the p elements of largest weight in the optimal solution. Using contraction (Lemma 1.2) we remove these elements from both matroids, and using deletion (Lemma 1.1) we as well remove all elements that have a larger weight than any of the contracted elements. We decrease the budget by an amount equal to the cost of the guessed elements. Finally we apply the above algorithm and we add back the guessed elements to the solution. The final solution will have weight at least $\text{opt} - 2w'_{\max}$, where $w'_{\max}$ is the largest weight of the elements that remained after the guessing step. Since $\text{opt} \geq (2/\varepsilon)w'_{\max}$, we obtain a solution of weight at least $(1-\varepsilon)\text{opt}$. The running time of the algorithm above is $O(m^{O(1/\varepsilon)})$. ∎

5   Conclusions and Open Problems

In this paper we presented PTASs for the budgeted version of two fundamental problems: maximum-weight matching and maximum-weight matroid intersection. Our approach resembles similar results in the literature, based on Lagrangian relaxation and patching. However, standard patching techniques seem not to be able to provide good approximation algorithms for our problems. This is not surprising, since the combinatorial structure of the underlying unbudgeted problems in our case is more complicated. In order to overcome this difficulty, we designed a novel patching technique. One crucial ingredient in our patching procedure was the solution to an old combinatorial puzzle.

There are several problems that we left open. One natural question is whether we can apply our patching technique to other budgeted problems. Apparently, the main ingredient we need is that the difference between two adjacent vertices in the polytope of the solutions to the corresponding unbudgeted problem can be characterized by a proper alternating path or cycle. This deserves further investigation.

Another natural question is whether there are FPTASs for the problems considered. We conjecture that budgeted matching is not strongly NP-hard. However, finding an FPTAS for that problem might be a very difficult task. In fact, for polynomial weights and costs, the budgeted matching problem is equivalent to the *exact perfect matching problem* (see Appendix): Given an undirected graph $G = (V,E)$, edge weights $w : E \rightarrow \mathbb{Q}$, and a parameter $W \in \mathbb{Q}$, find a perfect matching of weight exactly W, if any. This problem was first posed by Papadimitriou and Yannakakis [17]. For polynomial weights, the problem admits a polynomial-time Monte Carlo algorithm [3, 13]. Hence, it is very unlikely that exact perfect matching with polynomial weights is NP-hard (which would imply RP=NP). However, after 25 years, the problem of finding a deterministic algorithm to solve this problem is still open.

Finally, an interesting open problem is whether our approach can be extended to the case of multiple budget constraints. The difficulty here is that the Gasoline Lemma alone seems not able to fill in the cost-budget gap for several budget constraints at the same time

References

[1] V. Aggarwal, Y. P. Aneja, and K. P. K. Nair. Minimal spanning tree subject to a side constraint. *Computers & Operations Research*, 9(4):287–296, 1982.

[2] J. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394, 1989.

[3] P. Camerini, G. Galbiati, and F. Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13:258–273, 1992.

[4] A. Frank and É. Tardos. Generalized polymatroids and submodular flows. *Mathematical Programming*, 42:489–563, 1988.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[6] M. Guignard and M. B. Rosenwein. An application of lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. *Networks*, 20(3):345–359, 1990.

[7] R. Hassin and A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM J. Comput.*, 33(2):261–268, 2004.

[8] S.-P. Hong, S.-J. Chung, and B. H. Park. A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. *Oper. Res. Lett.*, 32(3):233–239, 2004.

[9] S. Iwata. On matroid intersection adjacency. *Discrete Mathematics*, 242:277–281, 2002.

[10] L. Lovász. *Combinatorial Problems and Exercises*. North-Holland, 1979.

[11] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. H. III. Bicriteria network design problems. In *Automata, Languages and Programming*, pages 487–498, 1995.

[12] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, 1979.

[13] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[14] J. Naor, H. Shachnai, and T. Tamir. Personal communication, 2007.

[15] J. Naor, H. Shachnai, and T. Tamir. Real-time scheduling with a budget. *Algorithmica*, 47(3):343–364, 2007.

[16] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, 1988.

[17] C. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29:285–309, 1982.

[18] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem (extended abstract). In *SWAT*, pages 66–75, 1996.

[19] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[20] A. Schrijver. *Combinatorial Optimization. Polyhedra and efficiency.*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003.

[21] D. B. Shmoys and É. Tardos. Scheduling unrelated machines with costs. In *SODA*, pages 448–454, 1993.

Appendix: Exact Perfect Matching and Budgeted Matching

Here we discuss the relation between the budgeted matching problem and the exact perfect matching problem. In particular, we show that the two problems are equivalent for polynomial weights and costs.

Recall that the *exact perfect matching problem* is the problem of determining whether an undirected graph $G = (V, E)$, with edge weights $w : E \to \mathbb{Q}$, admits a perfect matching of weight exactly $W \in \mathbb{Q}$. Let moreover the *budgeted perfect matching problem* be the variant of the budgeted matching problem where we also require that the matching returned is perfect.

Lemma 8. *For polynomial weights and costs, the following problems are polynomially reducible:*

    (a) exact perfect matching;   (b) budgeted perfect matching;   (c) budgeted matching.

Proof. Without loss of generality we will assume that weights and costs are nonnegative integers. As usual, $w_{\max}$ and $c_{\max}$ denote the largest weight and cost, respectively.

(a) $\Rightarrow$ (b). Let (G, w, W) be the input exact perfect matching instance. Solve the budgeted perfect matching instance (G, w, c, B), where $B = W$ and $c(e) = w(e)$ for every edge e. If the solution returned has weight smaller than $B = W$, the original problem is infeasible. Otherwise, the solution computed is a perfect matching of G of weight W.

(c) $\Rightarrow$ (a) Let (G, w, c, B) be the input budgeted matching instance. For two given W^* and C^*, consider the exact perfect matching instance (G, w', W'), where $W' = (n/2 + 1)c_{\max}W^* + B^*$ and $w'(e) = (n/2 + 1)c_{\max}w(e) + c(e)$ for every edge e. Problem (G, w', W') is feasible if and only if there is a matching of weight W^* and cost B^* in the original problem. By trying all the (polynomially many) possible values for W^* and B^*, one obtains the desired solution to the original problem.

(b) $\Rightarrow$ (c) Let (G, w, c, B) be the input budgeted perfect matching instance. Consider the budgeted matching instance (G, w', c, B), where $w'(e) = w(e) + (n/2 + 1)w_{\max}$ for every edge e. The original problem is feasible if and only the maximum matching M^* of the new problem contains $n/2$ edges. In that case, M^* is a maximum perfect matching. $\square$